Narrative Ontology

Nicola Gugole
Mat. 619625

A project presented for the
Semantic Web exam

University of Pisa
A.Y. 2020/2021

Q8. Identify two different assertions that would make the ontology inconsistent.

Various assertions can make the ontology inconsistent, for example having two different individuals with the same key or giving to an individual two class types which are disjoint with each other or even by going against a cardinality restriction.

The first example is based on going against having *key uniqueness* for the individuals of a class. In the case below two different books are asserted to have the same ISBN, where *:hasISBN* is defined to be the key property of class *Book*. Having the same key creates a clash, which could be avoided by giving different ISBNs or alternatively not ensuring the inequivalence between *:book1* and *:book2* (this would end up in the reasoner assuming the two books are actually the same).

```
:book1 rdf:type owl:NamedIndividual ,
               :Book ;
       :hasISBN "978-3-598-21545-2"^^xsd:string .
:book2 rdf:type owl:NamedIndividual ,
               :Book ;
       :hasISBN "978-3-598-21545-2"^^xsd:string .
[ rdf:type owl:AllDifferent ;
  owl:distinctMembers ( :book1
                        :book2
                      )
] .
```

The second example is based on going against a cardinality restriction for a specific property. Having the cardinality restriction on *Book*:
```
hasISBN exactly 1 xsd:string
```
A *Book* has to follow the constraint of having exactly a single ISBN specified by the Data Property *:hasISBN*. Having two different ISBNs for a single *Book* creates an inconsistency.

```
:book rdf:type owl:NamedIndividual ,
              :Book ;
      :hasISBN "978-3-598-21545-2"^^xsd:string ,
               "978-3-598-22432-5"^^xsd:string .
```

Q9. Define the complex role inclusion axiom capturing the fact that if a narrator creates a narrative that is reported in a book that is published by a publisher, then the narrator has a contract with that publisher.

The solution axiom is the following, given:

```
Vc = {:Narrator, :Narrative, :Book, :Publisher}
Vop = {:creates, :reportedBy, :hasPublisher, :hasContractWith}
SubObjectPropertyOf(
        ObjectPropertyChain(
                :creates
                :reportedBy
                :hasPublisher
        )
        :hasContractWith
)
```

The same chain was represented in *Protégé* with the following mathematical composition:

```
(creates o reportedBy o hasPublisher) --> hasContractWith
```

Q10. Verify if the created ontology (including the complex role inclusion axiom defined in Q9) satisfies the global restrictions on the axioms of an OWL 2 DL ontology.

Global Restrictions on OWL 2 DL ontology axioms are extremely important to retain decidability, there are 5 different constraints which need to be satisfied:

- **Restriction on owl:topDataProperty**: satisfied because the ontology does not include *SubDataPropertyOf* axioms where *owl:topDataProperty* occurs as the first argument.

- **Restrictions on Datatypes**: satisfied since the ontology datatypes come exclusively from the *OWL 2 datatype map*, no new datatypes were defined.

- **Restriction on Simple Roles**: satisfied because no **composite object property** is used in an axiom of the forbidden kinds. To be more precise this depends on how the axiom 9 (*"The narrator has at least one contract with a publisher."*) is implemented since *:hasContractWith* is the only composite object property on which a restriction is imposed. In fact axiom 9 can be stated in two different ways:

  - using *ObjectSomeValuesFrom*, which is not part of the forbidden kinds presented in class and therefore keeps the Global Restrictions satisfied.
  - using *ObjectMinCardinality*, which is actually part of the forbidden kinds presented in class, leading to a clash with the Global Restrictions.

  Anyway the choice for the ontology fell on *ObjectSomeValuesFrom* as in this case the use of *some* fits better and has the exact same semantic meaning of *min 1*. Nevertheless it is interesting to notice how different reasoners change behaviour with respect to this particular case: *HermiT* does not complain when a composite object property is contained into a *min 1 Cardinality Restriction* while for example *Pellet* complains and refuses to reason. A possible explanation for the behaviour of *HermiT* is that it understands it is the equivalent to the use of *some*.

- **Restriction on the Property Hierarchy**: satisfied since the only two *property chains* in the ontology are not cyclically defined.

- **Restrictions on Anonymous Individuals**: satisfied since no *anonymous individuals* are in the ontology.

Q11. Queries in *SPARQL*.

Being perfectly aware of the possibility of using *a* instead of *rdf:type* thanks to *TURTLE*, the use of *rdf:type* in the queries listed below is purely a subjective preference:

```
prefix : <http://www.semanticweb.org/narrativeExam#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?realLocation (COUNT(?event) AS ?eventNumber)
WHERE{
  ?realLocation rdf:type :RealLocation .
  ?event rdf:type :Event ;
         :occursIn ?realLocation .
  }
GROUP BY ?realLocation
```

Q11.1. Find how many events occurred in real locations, grouped by location.

```
prefix : <http://www.semanticweb.org/narrativeExam#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?book
WHERE{
  ?book rdf:type :Book .
  ?publisher rdf:type :Publisher ;
         :publishes ?book ;
         :hasID ?publisherId .
  FILTER(?publisherId < 5000)
  }
```

Q11.2. Find all the books with the ID of the publisher lower than 5000.

```
prefix : <http://www.semanticweb.org/narrativeExam#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?eventWithNoHuman
WHERE{
  ?eventWithNoHuman rdf:type :Event .
  FILTER NOT EXISTS {
    ?character :isCharacterOf ?eventWithNoHuman ;
               rdf:type :Human .
  }
}
```

Q11.3. Find all the events that do not have any human participants.

```
prefix : <http://www.semanticweb.org/narrativeExam#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?title (COUNT(?narrative) AS ?numberOfNarratives) ?isbn ?publisher
WHERE{
  ?book rdf:type :Book;
        :hasISBN ?isbn;
        :hasTitle ?title;
        :hasPublisher ?publisher;
        :reports ?narrative .
  ?publisher rdf:type :Publisher .
  ?narrative rdf:type :Narrative .
}
GROUP BY ?book ?title ?isbn ?publisher
```

Q11.4. Find the number of the narratives that are published in a book, along with the title of the book, the ISBN code of the book and the publisher of the book.

```
prefix : <http://www.semanticweb.org/narrativeExam#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?event
WHERE{
  ?event rdf:type :Event .
  {
    ?human rdf:type :Human ;
           :isCharacterOf ?event .
  }
  UNION
  {
    ?realLocation rdf:type :RealLocation ;
                  :isPlaceOf ?event .
  }
}
```

Q11.5. Find all the distinct events that have a human participant or occur in a real location.